# **EAST Search History**

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	365	717/141.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/23 13:56
S2	190	717/144.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/01/23 13:56
S3	88	717/157.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR -	OFF	2007/01/23 13:57
S4	116	inlin\$3 and call\$3 and (affinity or dependence) near3 (graph\$3 or node or tree or model\$3) and weight\$3 and (edge or arc or link\$3 or node)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 14:52
S5	13	(generat\$3 or creat\$3 or reorder\$3 or restructur\$3 ) near5 call\$3 same inlin\$3 same (depend\$4 )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 14:06
S6	19	(generat\$3 or creat\$3 or reorder\$3 or restructur\$3 or determin\$5 ) near5 call\$3 same inlin\$3 same (depend\$4 )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON .	2007/01/23 14:07
S7	11	(generat\$3 or creat\$3 or reorder\$3 or restructur\$3 or determin\$5 ) near5 call\$3 same inlin\$3 same (depend\$4 ) and performance	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 14:08
S8	254	inlin\$3 and call\$3 and (affinity or depen\$5 or dominator ) near3 (graph\$3 or node or tree or model\$3) and weight\$3 and (edge or arc or link\$3 or node)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 16:19
S9	6	inlin\$3 and call\$3 and (affinity or depen\$5 or dominator ) near3 (graph\$3 or node or tree or model\$3) and weight\$3 and (edge or arc or link\$3 or node) and (elimin\$5 near3 overhead)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR .	ON	2007/01/23 15:29

# **EAST Search History**

S10	565	S1 S2 S3 and (inlin\$3 or "in-lining" )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON .	2007/01/23 15:41
S11	559	S1 S2 S3 and (inlin\$3 or "in-lining" ) and (graph\$3 )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 15:41
S12	555	S1 S2 S3 and (inlin\$3 or "in-lining" ) and (graph\$3 near3 call\$3 )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 15:30
S13	550	S1 S2 S3 and (inlin\$3 or "in-lining" ) and (graph\$3 near3 call\$3 ) and weight\$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 15:30
S14	548	S1 S2 S3 and (inlin\$3 or "in-lining" ) and (graph\$3 near3 call\$3 ) and weight\$3 and ((affinity or dependen\$4) near3 graph\$3 )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 15:33
S15	547	S1 S2 S3 and (inlin\$3 or "in-lining" ) same (graph\$3 near3 call\$3 ) and weight\$3 and ((affinity or dependen\$4) near3 graph\$3 )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR ·	ON	2007/01/23 15:33
S16	546	S1 S2 S3 and (inlin\$3 or "in-lining" ) same (graph\$3 near3 call\$3 ) and weight\$3 and ((affinity or dependen\$4) near3 graph\$3 ) and (opened or active) adj files	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 15:34
S17	546	S1 S2 S3 and (inlin\$3 or "in-lining" ) same (graph\$3 near3 call\$3 ) and weight\$3 same ((affinity or dependen\$4) near3 graph\$3 ) and (opened or active) adj files	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR .	ON	2007/01/23 15:35
S18	0	S3 and (analyze or analysis or analyzing) same (inlin\$3 or "in-lining") same (graph\$3 near3 call\$3) and weight\$3 same ((affinity or dependen\$4) near3 graph\$3) and (opened or active) adj files	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 15:39

# **EAST Search History**

S19	1	S2 and (analyze or analysis or analyzing) same (inlin\$3 or "in-lining" ) same (graph\$3 near3 call\$3 ) and weight\$3 same ((affinity or dependen\$4) near3 graph\$3 ) and (opened or active) adj files	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 15:39
S20	0	S3 and (analyze or analysis or analyzing) same (inlin\$3 or "in-lining" ) same (graph\$3 near3 call\$3 ) and weight\$3 same ((affinity or dependen\$4) near3 graph\$3 )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 15:40
S21		717/15?.ccls. and (analyze or analysis or analyzing) same (inlin\$3 or "in-lining" ) same (graph\$3 near3 call\$3 ) and weight\$3 same ((affinity or dependen\$4) near3 graph\$3 )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 15:41
S22	79	(S1 S2 S3) and (inlin\$3 or "in-lining")	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 15:41
S23	45	(S1 S2 S3) and (inlin\$3 or "in-lining" ) and (graph\$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR .	ON	2007/01/23 16:01
S24	14	("5428793" "555417" "5920723" "6195793" "7028293" ).pn. or "20040064809"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 16:03
S25 .		("5428793" "5555417" "5920723" "6195793" "7028293" ).pn. or "20040064809"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 16:03
S26	2	"20050097527"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/23 16:19



Subscribe (Full Service) Register (Limited Sel Search: • The ACM Digital Library +code +expansion +I/O

Feedback Report a problem

Published since January 1980 and Published before October 2003

Terms used code expansion I/O

Sort results by

relevance

Save results to a Binder

Try an Advanc

Search Tips

Try this search

Display results

expanded form

□ Open results in a new window

Results 1 - 20 of 200

Result page: 1 2 3 4 5 6 7 8 9 10

Best 200 shown

Ro

ne

]

1 Using a lookahead window in a compaction-based parallelizing compiler

Toshio Nakatani, Kemal Ebcio□lu

January 1991 ACM SIGMICRO Newsletter, Volume 22 Issue 1

**Publisher:** ACM Press

Full text available: pdf(969.83 Additional Information: full citation, abst KB) index terms

Lookahead is a common technique for high performance uniprocessor de however, hardware lookahead window is too small to exploit instruction run time, while compaction-based parallelizing compilers must suffer fro exponential code explosion at compile time. In this paper, we propose a. method, which allows inter-basic block code motions within the prespec operations, called software lookahead window, ...

2 Using a lookahead window in a compaction-based parallelizing compiler Toshio Nakatani, Kemal Ebcioğlu

November 1990 Proceedings of the 23rd annual workshop and symposi Microprogramming and microarchitecture MICRO 23

Publisher: IEEE Computer Society Press

MB)

Full text available: pdf(1.11 Additional Information: full citation, abst citings

Lookahead is a common technique for high performance uniprocessor de however, hardware lookahead window is too small to exploit instruction run time, while compaction-based parallelizing compilers must suffer from exponential code explosion at compile time. In this paper, we propose a method, which allows inter-basic block code motions within the prespect operations, called software lo ...

3 Automatic microcode generation for horizontally microprogrammed proces

Robert J. Sheraga, John L. Gieser

December 1981 ACM SIGMICRO Newsletter, Proceedings of the 14th on Microprogramming MICRO 14, Volume 12 Issue 4

Publisher: IEEE Press, ACM Press

Full text available: pdf(1.22 Additional Information: full citation, abst citings, index ten

A procedure is described which permits applications problems coded in Language to be compiled to microcode for horizontally microprogramme experimental language has been designed which is suitable for expressin oriented problems for such processors in a distributed processing environ programs are compiled first to a machine independent intermediate language machine dependent form consisting of elementary microoperatio ...

4 Maximal static expansion

Denis Barthou, Albert Cohen, Jean-François Collard
January 1998 Proceedings of the 25th ACM SIGPLAN-SIGACT sympo
of programming languages POPL '98

**Publisher:** ACM Press

Full text available: pdf(1.19 Additional Information: full citation, reference MB)

Additional Information: full citation, reference makes index terms

Keywords: expansion of data structure, privatization, single assignment

- 5 Views on transportability of Lisp and Lisp-based systems
- Richard J. Fateman
  August 1981 Proceedings of the fourth ACM symposium on Symbolic 2

## computation SYMSAC '81

**Publisher:** ACM Press

Full text available: pdf(489.58 Additional Information: full citation, abst KB) citings, index ten

The availability of new large-address-space computers has provided us a examine techniques for transferring programming systems, and in particuto new computers. We contrast two approaches: designing and building a implementation of Lisp, and (re)writing the system in a "portable" progr ('C'). Our conclusion is that the latter approach may very well be better.

6 <u>Discrete event simulation using PL/I based general and special purpose sin</u> Walter C. Metz

January 1981 Proceedings of the 13th conference on Winter simulation '81

**Publisher: IEEE Press** 

Full text available: pdf(650.17 Additional Information: full citation, abst KB) citings, index ten

This paper describes the architecture and language features of a simulation developed using a new IBM discrete event simulation package based on contains implementations of both the GPSS and SIMPL/I simulation languages the capability for a model developer to create special planguages tailored to his unique simulation application. The model describing simulates a retail or supermarket store point-of-s ...

7 Multilingual text processing in a two-byte code

Lloyd B. Anderson

July 1984 Proceedings of the 22nd annual meeting on Association for C Linguistics, Proceedings of the 10th international conference Computational linguistics

Publisher: Association for Computational Linguistics

Full text available: Dpdf(368.42

KB) 🖣 Publisher

Additional Information: full citation, abst

<u>Site</u>

National and international standards committees are now discussing a tw multilingual information processing. This provides for 65,536 separate c

codes, enough to make permanent code assignments for all the character alphabets of the world, and also to include Chinese/Japanese characters. the kinds of flexibility required to handle both Roman and non-Roman a crucial to separate information units (codes) from gr ...

8 Compiler code transformations for superscalar-based high performance sys S. A. Mahlke, W. Y. Chen, J. C. Gyllenhaal, W.-M. W. Hwu December 1992 Proceedings of the 1992 ACM/IEEE conference on Sup Supercomputing '92

Publisher: IEEE Computer Society Press

Full text available: pdf(1.05 Additional Information: full citation, reference MB)

Additional Information: full citation, reference makes index terms

9 Interactive conversion of sequential to multitasking FORTRAN

Kevin Smith, Bill Appelbe.

June 1989 Proceedings of the 3rd international conference on Supercor Publisher: ACM Press

Full text available: pdf(972.31 Additional Information: full citation, abst KB) citings, index ten

Fully automated compilation of sequential Fortran to efficient multitaski impractical; tools need to be developed to aid users in interactively conv multitasking Fortran. This paper reports on experience using an interacti Assistant Tool (PAT) to convert sequential Fortran applications (ranging benchmarks to large application programs) to Cray microtasking Fortran advantages and limitations of interactive paralleliz ...

10 HARE: an optimizing portable compiler for Scheme

Dan Teodosiu

January 1991 ACM SIGPLAN Notices, Volume 26 Issue 1

**Publisher:** ACM Press

Full text available: pdf(872.48 KB) Additional Information: full citation, abst

A highly optimizing Scheme compiler called HARE is presented. A com optimization techniques allows for the generation of very efficient code. the compiler has been achieved through the use of a virtual machine as a

generation. The compiler will be used as a test-bed for fine-tuning the in symbolic architecture, the S-Machine.

11 Software pipelining loops with conditional branches

Mark G. Stoodley, Corinna G. Lee

December 1996 Proceedings of the 29th annual ACM/IEEE internation Microarchitecture MICRO 29

**Publisher: IEEE Computer Society** 

Full text available: Additional Information: <u>full citation</u>, <u>abst</u> <u>ocitings</u>, <u>index ten</u>

Software pipelining is an aggressive scheduling technique that generates loops and is particularly effective for VLIW architectures. Few software algorithms, however, are able to efficiently schedule loops that contain c We have developed an algorithm we call All Paths Pipelining (APP) that shortcoming of software pipelining. APP is designed to achieve optimal performance for any run of iterations while providing ef ...

- 12 Design decisions influencing the microarchitecture for a Prolog machine
- T. P. Dobry, Y. N. Patt, A. M. Despain

December 1984 ACM SIGMICRO Newsletter, Proceedings of the 17th on Microprogramming MICRO 17, Volume 15 Issue 4

Publisher: IEEE Press, ACM Press

Full text available: pdf(1.27 Additional Information: full citation, abst citings, index ten

The PLM-1 is the first step in the hardware implementation of a heteroge processor for logic programming. This paper describes its ISP architectu detail some of the design decisions relative to its microarchitecture.

- 13 Trace-driven memory simulation: a survey
- Richard A. Uhlig, Trevor N. Mudge

June 1997 ACM Computing Surveys (CSUR), Volume 29 Issue 2

**Publisher:** ACM Press

Full text available: pdf(636.11 Additional Information: full citation, abst KB) citings, index ten

As the gap between processor and memory speeds continues to widen, n evaluating memory system designs before they are implemented in hards

increasingly important. One such method, trace-driven memory simulations subject of intense interest among researchers and has, as a result, enjoyed and substantial improvements during the past decade. This article survey developments by establishing criteria for evaluating trac ...

**Keywords**: TLBs, caches, memory management, memory simulation, trasimulation

14 <u>Techniques for efficient inline tracing on a shared-memory multiprocessor</u>

S. J. Eggers, David R. Keppel, Eric J. Koldinger, Henry M. Levy

April 1990 ACM SIGMETRICS Performance Evaluation Review, Pro 1990 ACM SIGMETRICS conference on Measurement and computer systems SIGMETRICS '90, Volume 18 Issue 1

**Publisher:** ACM Press

Full text available: pdf(1.12 Additional Information: full citation, abst citings, index ten

While much current research concerns multiprocessor design, few traces programs are available for analyzing the effect of design trade-offs. Exis methods have serious drawbacks: trap-driven methods often slow down by more than 1000 times, significantly perturbing program behavior; mic modification is faster, but the technique is neither general nor portable. It a new tool, called MPTRACE, for collecting tr...

15 Implementing functional languages in the Categorical Abstract Machine

Michel Mauny, Ascánder Suárez

August 1986 Proceedings of the 1986 ACM conference on LISP and ful programming LFP '86

**Publisher:** ACM Press

Full text available: pdf(687.85 Additional Information: full citation, refe

16 A Fortran preprocessor for the large program environment

Neal R. Wagner
December 1980 ACM SIGPLAN Notices, Volume 15 Issue 12

**Publisher:** ACM Press

Full text available: pdf(902.71 KB) Additional Information: full citation, abst

The use of a preprocessor to aid structured programming in Fortran has I discussed. This article considers a design philosophy which is especially large program development and maintenance. The design is distinguishe the form of the original source program in the standard Fortran output by A specific implementation is described.

17 A survey of resource allocation methods in optimizing microcode compiler

Robert A. Mueller, Michael R. Duda, Stephen M. O'Haire

December 1984 ACM SIGMICRO Newsletter, Proceedings of the 17th on Microprogramming MICRO 17, Volume 15 Issue 4

Publisher: IEEE Press, ACM Press

Full text available: pdf(887.10 Additional Information: full citation, abst KB) index terms

This paper surveys results reported on resource allocation in optimizing compilers. Resource allocation is the phase of microcode generation that operators of program text to machine registers and functional units. The results on resource allocation in optimizing microcode compilers were reand subsequent results were reported by Kim and Tan and by Ma and Le each of these methods, focusing on th ...

18 The Java syntactic extender (JSE)

Jonthan Bachrach, Keith Playford

October 2001 ACM SIGPLAN Notices, Proceedings of the 16th ACM stransference on Object oriented programming, systems, lail applications OOPSLA '01, Volume 36 Issue 11

**Publisher:** ACM Press

Full text available: pdf(198.11 Additional Information: full citation, abst KB) citings, index ten

The ability to extend a language with new syntactic forms is a powerful if flexible macro system allows programmers to build from a common base language designed specifically for their problem domain. However, macintegrated, capable, and at the same time simple enough to be widely use to the Lisp family of languages to date. In this paper we introduce a mac

the Java Syntactic Extender (JSE), with the superio ...

### 19 An efficient variable-cost maze router

Robert K. Korn

January 1982 Proceedings of the 19th conference on Design automation Publisher: IEEE Press

Full text available: pdf(554.89 Additional Information: full citation, abst KB) citings, index ten

A variable cost maze router is described. The router is substantially faste maze routers and also provides a flexibility which is valuable in a variety particularly well suited for use on multiple layer routing surfaces in which have primary wire directions which are perpendicular to each other. The incorporated as a final phase into both a circuit board routing system and router. Experience with these systems ...

### 20 MIL primitives for querying a fragmented world

Peter A. Boncz, Martin L. Kersten

October 1999 The VLDB Journal — The International Journal on Ver Bases, Volume 8 Issue 2

Publisher: Springer-Verlag New York, Inc.

Full text available: Pdf(261.36 Additional Information: full citation, abst KB)

KB)

terms

In query-intensive database application areas, like decision support and ce that use vertical fragmentation have a significant performance advantage relational or object oriented applications on top of such a fragmented dat yet powerful intermediate language is needed. This problem has been sue Monet, a modern extensible database kernel developed by our group. We design choices made in the Monet interprete ...

Keywords: Database systems, Main-memory techniques, Query languag optimization, Vertical fragmentation

Results 1 - 20 of 200

Result page: 1 2 3 4 5 6 7 8 9 1

The ACM Portal is published by the Association for Computing Machinery

ACM, Inc.

Terms of Usage Privacy Policy Code of Ethics Contact

Useful downloads: Adobe Acrobat QuickTime Windows Med Player



Subscribe (Full Service) Register (Limited Ser **Search:** • The ACM Digital Library +inlin\* +call +graph

Feedback Report a problem

Published since January 1980 and Published before October 2003

Ş

Terms used inlin call graph

Sort results relevance by

Save results to a Binder Search Tips

Try an Advanc Try this search

Display results

Š expanded form

□ Open results in a new

window

Results 1 - 20 of 200

Result page: 1 2 3 4 5 6 7 8 9 10

Best 200 shown

 $n\epsilon$ R

1 Inline function expansion for compiling C programs

P. P. Chang, W.-W. Hwu

June 1989 ACM SIGPLAN Notices, Proceedings of the ACM SIGPLA on Programming language design and implementation PLD] Issue 7

**Publisher:** ACM Press

Full text available: Dpdf(1.14 MB)

Additional Information: full citation, abst citings, index ten

Inline function expansion replaces a function call with the function body inline function expansion, programs can be constructed with many small complexity and then rely on the compilation to eliminate most of the fun Therefore, inline expansion serves a tool for satisfying two conflicting go complexity of the program development and minimizing the function cal program execution. A simple inline expansion procedur ...

2 A comparative study of static and profile-based heuristics for inlining

Matthew Arnold, Stephen Fink, Vivek Sarkar, Peter F. Sweeney January 2000 ACM SIGPLAN Notices, Proceedings of the ACM SIGP. Dynamic and adaptive compilation and optimization DYI Volume 35 Issue 7

**Publisher:** ACM Press

Full text available: pdf(1.13 Additional Information: full citation, abst citings, index ten

In this paper, we present a comparative study of static and profile-based inlining. Our motivation for this study is to use the results to design the talgorithm that we can for the Jalapeño dynamic optimizing compiler for well-known approximation algorithm for the KNAPSACK problem as a algorithm" for the inlining heuristics studied in this paper. We present pe for an implementation of these inlinin ...

3 Practical virtual method call resolution for Java

Vijay Sundaresan, Laurie Hendren, Chrislain Razafimahefa, Raja Vallée-R Etienne Gagnon, Charles Godin

October 2000 ACM SIGPLAN Notices, Proceedings of the 15th ACM sconference on Object-oriented programming, systems, la applications OOPSLA '00, Volume 35 Issue 10

**Publisher:** ACM Press

Full text available: Pdf(323.98 Additional Information: full citation, abst KB)

KB) citings, index ten

This paper addresses the problem of resolving virtual method and interfa bytecode. The main focus is on a new practical technique that can be use applications. Our fundamental design goal was to develop a technique th with only one iteration, and thus scales linearly with the size of the programe time providing more accurate results than two popular existing line hierarchy analysis and rapid type an ...

- 4 On the conversion of indirect to direct recursion
- Owen Kaser, C. R. Ramakrishnan, Shaunak Pawagi

March 1993 ACM Letters on Programming Languages and Systems (L 2 Issue 1-4

**Publisher:** ACM Press

Full text available: pdf(929.68 Additional Information: full citation, abst KB) citings, index ten

Procedure inlining can be used to convert mutual recursion to direct recu use of optimization techniques that are most easily applied to directly rec in addition to the well-known benefits of inlining. We present tight (nece sufficient) conditions under which inlining can transform all mutual recurecursion, and those under which heuristics to eliminate mutual recursion. We also present a technique ...

Keywords: call graphs, inline substitution, mutual recursion, procedure:

5 <u>Dynamic Adaptive compilation: Adaptive online context-sensitive inlining</u> Kim Hazelwood, David Grove

March 2003 Proceedings of the international symposium on Code generoptimization: feedback-directed and runtime optimization

Publisher: IEEE Computer Society

Full text available: pdf(1.06 Additional Information: full citation, abst citings, index ten

As current trends in software development move toward more complex of programming, inlining has become a vital optimization that provides subsperformance improvements to C++ and Java programs. Yet, the aggression inlining algorithm must be carefully monitored to effectively balance per size. The state-of-the-art is to use profile information (associated with cainlining decisions. In the presence of virtual method calls, profile ...

- 6 Sealed calls in Java packages
- Ayal Zaks, Vitaly Feldman, Nava Aizikowitz

October 2000 ACM SIGPLAN Notices, Proceedings of the 15th ACM sconference on Object-oriented programming, systems, la applications OOPSLA '00, Volume 35 Issue 10

**Publisher:** ACM Press

Full text available: pdf(192.57 Additional Information: full citation, abst KB) citings, index ten

Determining the potential targets of virtual method invocations is essenti procedural optimizations of object-oriented programs. It is generally hard targets accurately. The problem is especially difficult for dynamic languages additional targets of virtual calls may appear at runtime. Current enable inter-procedural optimizations for dynamic languages, repeatedly optimizations at runtime. This paper addresses this ...

Keywords: Java, call devirtualization, call graph, class hierarchy graph,

analysis, method inlining, object-oriented programming, sealed package

7 Aggressive inlining

Andrew Ayers, Richard Schooler, Robert Gottlieb

May 1997 ACM SIGPLAN Notices, Proceedings of the ACM SIGPLA on Programming language design and implementation PLD Issue 5

**Publisher:** ACM Press

Full text available: Additional Information: full citation, abst · MB) citings, index ten

Existing research understates the benefits that can be obtained from inlin especially when guided by profile information. Our implementation of ir yields excellent results on average and very rarely lowers performance. \ results can be explained by a number of factors: inlining at the intermedi removes most technical restrictions on what can be inlined; the ability to and incorporate profile information enables ...

- 8 Automatic pool allocation for disjoint data structures
- Chris Lattner, Vikram Adve

June 2002 ACM SIGPLAN Notices, Proceedings of the 2002 workshop system performance MSP '02, Volume 38 Issue 2 supplement

**Publisher:** ACM Press

Full text available: pdf(1.48 Additional Information: full citation, abst MB) citings

This paper presents an analysis technique and a novel program transform enable powerful optimizations for entire linked data structures. The fully transformation converts ordinary programs to use pool (aka region) alloc based data structures. The transformation relies on an efficient link-time analysis to identify disjoint data structures in the program, to check whet structures are accessed in a type-safe manner, and to constru...

- 9 Partitioning sequential programs for CAD using a three-step approach
- Frank Vahid

July 2002 ACM Transactions on Design Automation of Electronic Syst Volume 7 Issue 3

**Publisher:** ACM Press

Full text available: pdf(147.12 Additional Information: full citation, abst KB) citings, index ten

Many computer-aided design problems involve solutions that require the large sequential program written in a language such as C or VHDL. Such improve design metrics such as performance, power, energy, size, input/even CAD tool run-time and memory requirements, by partitioning amoundules, hardware and software processors, or even among time-slices i computing devices. Previous partitioning approaches typically presel ...

**Keywords**: Partitioning, behavioral partitioning, functional partitioning, partitioning, system level partitioning

## 10 Flow-directed inlining

Suresh Jagannathan, Andrew Wright

May 1996 ACM SIGPLAN Notices, Proceedings of the ACM SIGPLA on Programming language design and implementation PLD Issue 5

Publisher: ACM Press

Full text available: Pdf(1.33 Additional Information: full citation, abst citings, index ten

A *flow-directed inlining* strategy uses information derived from control-specialize and inline procedures for functional and object-oriented langu control-flow analysis to identify candidate call sites, flow-directed inliniprocedures whose relationships to their call sites are not apparent. For in defined in other modules, passed as arguments, returned as values, or extructures can all be inlined. Flow-d ...

11 Online feedback-directed optimization of Java

Matthew Arnold, Michael Hind, Barbara G. Ryder
November 2002 ACM SIGPLAN Notices, Proceedings of the 17th ACN
conference on Object-oriented programming, systems,
applications OOPSLA '02, Volume 37 Issue 11

**Publisher:** ACM Press

Full text available: pdf(463.00 Additional Information: full citation, abst

KB)

citings, index ten

This paper describes the implementation of an online feedback-directed system. The system is fully automatic; it requires no prior (offline) profil previously developed low-overhead instrumentation sampling frameworl flow graph edge profiles. This profile information is used to drive severa optimizations, as well as a novel algorithm for performing feedback-dire graph node splitting. We empirically evaluate this syst ...

**Keywords**: adaptive optimization, dynamic optimization, online algorithmachines

12 Interprocedural conditional branch elimination

Rastislav Bodík, Rajiv Gupta, Mary Lou Soffa

May 1997 ACM SIGPLAN Notices, Proceedings of the ACM SIGPLA on Programming language design and implementation PLD Issue 5

**Publisher:** ACM Press

Full text available: Pdf(2.02 Additional Information: full citation, abst citings, index ten

The existence of statically detectable correlation among conditional bran elimination, an optimization that has a number of benefits. This paper pr determine whether an interprocedural execution path leading to a conditi along which the branch outcome is known at compile time, and then to e along this path through code restructuring. The technique consists of a dinterprocedural analysis that determines whethe ...

13 Unexpected side effects of inline substitution: a case study

Keith D. Cooper, Mary W. Hall, Linda Torczon

March 1992 ACM Letters on Programming Languages and Systems (L 1 Issue 1

**Publisher:** ACM Press

Full text available: Pdf(740.92 Additional Information: full citation, abst KB) citings, index ten

The structure of a program can encode implicit information that changes speed of the generated code. Interprocedural transformations like inlining

information; using interprocedural data-flow information as a basis for o have the same effect. In the course of a study on inline substitution with FORTRAN compilers, we encountered unexpected performance problen programs. This paper describes the specific ...

Keywords: inline substitution, interprocedural analysis, interprocedural

14 The Jalapeño dynamic optimizing compiler for Java

Michael G. Burke, Jong-Deok Choi, Stephen Fink, David Grove, Michael Mauricio J. Serrano, V. C. Sreedhar, Harini Srinivasan, John Whaley June 1999 Proceedings of the ACM 1999 conference on Java Grande J. Publisher: ACM Press

Full text available: pdf(1.34 Additional Information: full citation, reference MB)

Additional Information: full citation, reference index terms

15 Polymorphic splitting: an effective polyvariant flow analysis

Andrew K. Wright, Suresh Jagannathan
January 1998 ACM Transactions on Programming Languages and Syst
Volume 20 Issue 1

**Publisher:** ACM Press

Full text available: pdf(517.76 Additional Information: full citation, abst citings, index ten

This article describes a general-purpose program analysis that computes and data-flow information for higher-order, call-by-value languages. The novel form of polyvariance called polymorhic splitting that uses let-expr clues to gain precision. The information derived from the analysis is used run-time checks and to inline procedure. The analysis and optimizations to a suite of Scheme progra ...

Keywords: flow analysis, inlining, polyvariance, run-time checks

16 Gprof: A call graph execution profiler

Susan L. Graham, Peter B. Kessler, Marshall K. Mckusick

June 1982 ACM SIGPLAN Notices, Proceedings of the 1982 SIGPLAN Compiler construction SIGPLAN '82, Volume 17 Issue 6

**Publisher:** ACM Press

Full text available: pdf(684.69 Additional Information: full citation, abst KB) citings, index ten

Large complex programs are composed of many small routines that impl for the routines that call them. To be useful, an execution profiler must a time in a way that is significant for the logical structure of a program as textual decomposition. This data must then be displayed to the user in a informative way. The gprof profiler accounts for the running time of call running time of the routines ...

17 Static conflict analysis for multi-threaded object-oriented programs

Christoph von Praun, Thomas R. Gross

May 2003 ACM SIGPLAN Notices, Proceedings of the ACM SIGPLA on Programming language design and implementation PLD Issue 5

**Publisher:** ACM Press

Full text available: pdf(674.11 Additional Information: full citation, abst KB) citings, index ten

A compiler for multi-threaded object-oriented programs needs information of objects for a variety of reasons: to implement optimizations, to issue variety instrumentation to detect access violations that occur at runtime. An Object (OUG) statically captures accesses from different threads to objects. An Heap Shape Graph (HSG), which is a compile-time abstraction for runting and their reference relations (edges). An OUG specifie ...

**Keywords**: heap shape graph, object use graph, program analysis, race d representations for concurrent programs

18 An efficient register optimization algorithm for high-level synthesis from h

**behavioral** specifications

Ranga Vemuri, Srinivas Katkoori, Meenakshi Kaul, Jay Roy January 2002 ACM Transactions on Design Automation of Electronic S (TODAES), Volume 7 Issue 1

**Publisher:** ACM Press

Full text available: pdf(571.24 Additional Information: full citation, abst KB) citings, index ten

We address the problem of register optimization that arises during high-l modular hierarchical behavioral specifications. Register optimization is t grouping carriers such that each group can be safely allocated to a hardw register optimization by inline expansion involves flattening the module a heuristic register optimization procedure on the flattened description. A expansion yields a near-optimal number of ...

Keywords: Behavioral synthesis, hardware description languages, hierar specifications, high-level synthesis, lifecycle analysis, register optimizat.

19 A framework for call graph construction algorithms

David Grove, Craig Chambers

November 2001 ACM Transactions on Programming Languages and S (TOPLAS), Volume 23 Issue 6

**Publisher:** ACM Press

Full text available: pdf(1.36 Additional Information: full citation, abst citings, index ten

A large number of call graph construction algorithms for object-oriented languages have been proposed, each embodying different tradeoffs betw and call graph precision. In this article we present a unifying framework call graph construction algorithms and an empirical comparison of a replagorithms. We first present a general parameterized algorithm that enco known and novel call graph construction algorithms. W ...

Keywords: Call graph construction, control flow analysis, interprocedur

20 Practical extraction techniques for Java

Frank Tip, Peter F. Sweeney, Chris Laffra, Aldo Eisma, David Streeter November 2002 ACM Transactions on Programming Languages and S (TOPLAS), Volume 24 Issue 6

**Publisher:** ACM Press

Full text available: pdf(1.01 Additional Information: full citation, abst

MB)

citings, index ten

Reducing application size is important for software that is distributed via order to keep download times manageable, and in the domain of embedd applications are often stored in (Read-Only or Flash) memory. This pape extraction techniques such as the removal of unreachable methods and reinlining of method calls, and transformation of the class hierarchy for recisize. We implemented a number of extraction techniques in < ...

**Keywords**: Application extraction, call graph construction, class hierarc packaging, whole-program analysis

Results 1 - 20 of 200

Result page: 1 2 3 4 5 6 7 8 9 1

The ACM Portal is published by the Association for Computing Machinery ACM, Inc.

Terms of Usage Privacy Policy Code of Ethics Contact

Useful downloads: Adobe Acrobat QuickTime Windows Med
Player



Subscribe (Full Service) Register (Limited Ser • The ACM Digital Library Search: O The +inlin\* + reduc\* +I/O

Feedback Report a problem

Terms used inlin reduc I/O

Sort results by

relevance

Save results to a Binder 2 Search Tips

Try an Advanc Try this search

Display

results

expanded form

□ Open results in a new window

Results 1 - 20 of 200

Result page: 1 <u>2</u> <u>3</u> <u>4</u> <u>5</u> <u>6</u> <u>7</u> <u>8</u> <u>9</u> <u>10</u>

ne ·

Best 200 shown

R

1 Scalable High Performance Cross-Module Inlining

Dhruva R. Chakrabarti, Luis A. Lozano, Xinliang D. Li, Robert Hundt, Shi September 2004 Proceedings of the 13th International Conference on P Architectures and Compilation Techniques PACT '04

**Publisher:** IEEE Computer Society

Full text available: pdf(241.65 Additional Information: full citation, abst

Performing inlining of routines across file boundaries is known to yield s performance improvements. In this paper, we present a scalable cross-makes framework that reduces the compiler's memory footprint, file thrashing, time. Instead of using the call-site ordering generated by the analysis pha transformation phase dynamically produces a new inlining order depend: constraints of the system. We introduce dependences among ...

2 <u>Using annotations to reduce dynamic optimization time</u>

Chandra Krintz, Brad Calder

May 2001 ACM SIGPLAN Notices, Proceedings of the ACM SIGPLA on Programming language design and implementation PLD Issue 5

**Publisher:** ACM Press

Full text available: Dpdf(1.78

Additional Information: full citation, abst

MB)

citings, index ten

Dynamic compilation and optimization are widely used in heterogenous environments, in which an intermediate form of the code is compiled to resecution. An important trade off exists between the amount of time spen optimizing the program and the running time of the program. The time to optimizations can cause significant delays during execution and also progains that result from more complex optimization.

- 3 Compiler analysis and optimization: Providing time- and space- efficient p
- asynchronous software thread integration

Vasanth Asokan, Alexander G. Dean

September 2004 Proceedings of the 2004 international conference on Coarchitecture, and synthesis for embedded systems CAS

**Publisher:** ACM Press

Full text available: pdf(289.56 Additional Information: full citation, abst KB) citings, index ten

Asynchronous Software Thread Integration (ASTI) provides fine-grain c time threads by statically scheduling (integrating) code from primary thr threads, reducing the context switching needed and allowing recovery of time. Unlike STI, ASTI allows asynchronous thread progress. Current AS not support procedure calls in the secondary thread because they lead to during static scheduling. ASTI requires knowing the sec ...

Keywords: asynchronous software thread integration, fine-grain concurs software migration, software-implemented communication protocol cont

- 4 An evaluation of automatic object inline allocation techniques
- Julian Dolby, Andrew A. Chien

October 1998 ACM SIGPLAN Notices, Proceedings of the 13th ACM sconference on Object-oriented programming, systems, la applications OOPSLA '98, Volume 33 Issue 10

**Publisher:** ACM Press

Full text available: pdf(2.26 Additional Information: full citation, abst citings, index ten

Object-oriented languages such as Java and Smalltalk provide a uniform model, allowing objects to be conveniently shared. If implemented direc

reference models can suffer in efficiency due to additional memory derememory management operations. Automatic *inline allocation* of child of objects can reduce overheads of heap-allocated pointer-referenced object compiler analyses to identify inlinable fields by t ...

5 Techniques for efficient inline tracing on a shared-memory multiprocessor

S. J. Eggers, David R. Keppel, Eric J. Koldinger, Henry M. Levy

April 1990 ACM SIGMETRICS Performance Evaluation Review, Pro 1990 ACM SIGMETRICS conference on Measurement and computer systems SIGMETRICS '90, Volume 18 Issue 1

**Publisher:** ACM Press

Full text available: pdf(1.12 Additional Information: full citation, abst citings, index ten

While much current research concerns multiprocessor design, few traces programs are available for analyzing the effect of design trade-offs. Exis methods have serious drawbacks: trap-driven methods often slow down by more than 1000 times, significantly perturbing program behavior; mic modification is faster, but the technique is neither general nor portable. I a new tool, called MPTRACE, for collecting tr ...

6 Exploiting the non-determinism and asynchrony of set iterators to reduce a

<u>latency</u>

David C. Steere

October 1997 ACM SIGOPS Operating Systems Review, Proceedings ACM symposium on Operating systems principles SOSP Issue 5

**Publisher:** ACM Press

Full text available: pdf(1.87 Additional Information: full citation, refe

MB) index terms

7 Controlling transmission order of inline objects for effective Web page put

Tadashi Nakano, Kaname Harumoto, Shinji Shimojo, Shojiro Nishio March 2000 Proceedings of the 2000 ACM symposium on Applied compact SAC '00

**Publisher:** ACM Press

Full text available: Pdf(571.96 Additional Information: full citation, reference KB)

KB)

index terms

Keywords: WWW, inline object, transmission order

8 Libraries and applications: Performance modeling and optimization of para

tensor contractions

Xiaoyang Gao, Swarup Kumar Sahoo, Chi-Chung Lam, J. Ramanujam, Qi Baumgartner, P. Sadayappan

June 2005 Proceedings of the tenth ACM SIGPLAN symposium on Pri practice of parallel programming PPoPP '05

**Publisher:** ACM Press

Full text available: pdf(136.72 Additional Information: full citation, abst KB) index terms

The Tensor Contraction Engine (TCE) is a domain-specific compiler for complex tensor contraction expressions arising in quantum chemistry ap electronic structure. This paper develops a performance model for tensor considering both disk I/O as well as inter-processor communication costs performance-model driven loop optimization for this domain. Experiment provided that demonstrate the accuracy and effectiveness of the mod ...

Keywords: compiler optimization, out-of-core algorithms, parallel algor modeling

9 Promises and reality: Performance measurements of a user-space DAFS se

workload

Samuel A. Fineberg, Don Wilson

August 2003 Proceedings of the ACM SIGCOMM workshop on Netwo convergence: experience, lessons, implications NICELI '0.

**Publisher:** ACM Press

Full text available: pdf(366.48 Additional Information: full citation, abst KB) index terms

We evaluate the performance of a user-space Direct Access File System

Oracle Disk Manager (ODM) client using two synthetic test codes as we database. Tests were run on 4-processor Intel Xeon-based systems running The systems were connected with ServerNet II, a Virtual Interface Archit compliant system area network. We compare the performance of DAFS/based I/O, measuring I/O bandwidth and latency. We also compare the r

Keywords: DAFS, Database, File Systems, I/O, Networks, Performance

10 The application development environment of the DECmpp 12000 massivel

computer—an introduction

Albert Lai, Eric Lo, Wing Cheong Man, Kam-Fai Wong

September 1993 **ACM SIGAPP Applied Computing Review**, Volume 1 **Publisher:** ACM Press

Full text available: Pdf(555.00 Additional Information: full citation, abst

This paper gives a brief introduction to the application development envi DECmpp 12000 Massively Parallel Computer. Specifically, the architect system and compilers are discussed.

11 Optimizing dynamically-dispatched calls with run-time type feedback

Urs Hölzle, David Ungar

June 1994 ACM SIGPLAN Notices, Proceedings of the ACM SIGPLA on Programming language design and implementation PLD Issue 6

**Publisher:** ACM Press

Full text available: pdf(1.39 Additional Information: full citation, reference MB)

Additional Information: full citation, reference index terms, reviews.

12 Procedure cloning: a transformation for improved system-level functional 1

Frank Vahid

January 1999 ACM Transactions on Design Automation of Electronic S (TODAES), Volume 4 Issue 1

**Publisher:** ACM Press

Full text available: pdf(227.98 Additional Information: full citation, abst KB) citings, index ten

Functional partitioning assigns the functions of a system's program-like system components, such as standard-software and custom-hardware prointroduce a new transformation, called procedure cloning, that significan functional partitioning results. The transformation creates a clone of a pr by a particular procedure caller, so the clone can be assigned to the caller in turn improves performance through reduced ...

**Keywords**: behavioral synthesis, embedded systems, functional partition hardware/software codesign, replication, system-level design, system-on transformations

## 13 SYZYGY - A Framework for Scalable Cross-Module IPO

Sungdo Moon, Xinliang D. Li, Robert Hundt, Dhruva R. Chakrabarti, Luis Srinivasan, Shin-Ming Liu

March 2004 Proceedings of the international symposium on Code generoptimization: feedback-directed and runtime optimization

Publisher: IEEE Computer Society

Full text available: pdf(198.14 KB) Additional Information: full citation, abst

Performing analysis across module boundaries for an entire program is in exploiting several runtime performance opportunities. However, due to so in existing full-program analysis frameworks, such performance opportunities opportunities to paying tremendous compile-timecosts. Alternative solutions, partial compilations or user assertions, are complicated or or or as a commercial applications are compiled to day with cross-module optimizations.

## 14 Space and time-efficient memory layout for multiple inheritance

Peter F. Sweeney, Joseph (Yossi) Gil

October 1999 ACM SIGPLAN Notices, Proceedings of the 14th ACM sconference on Object-oriented programming, systems, la applications OOPSLA '99, Volume 34 Issue 10

**Publisher:** ACM Press

Full text available: pdf(2.30 Additional Information: full citation, abst

### MB)

citings, index ten

Traditional implementations of multiple inheritance bring about not only terms of run-time but also a significant increase in object space. For exar compiler-generated fields in a certain object can be as large as quadratic subobjects. The problem of efficient object layout is compounded by the different semantics of multiple inheritance: shared, in which a base class distinct ...

## 15 Compiling C for vectorization, parallelization, and inline expansion

R. Allen, S. Johnson

June 1988 ACM SIGPLAN Notices, Proceedings of the ACM SIGPLA on Programming Language design and Implementation PLI Issue 7

**Publisher:** ACM Press

Full text available: pdf(1.09 Additional Information: full citation, abst citings, index ten

Practical implementations of real languages are often an excellent way o applicability of theoretical principles. Many stresses and strains arise fro practicalities, such as performance and standard compatibility, to theoret methods. These stresses and strains are valuable sources of new research as an oft-needed check on the egos of theoreticians. Two fertile areas that by implementations are

16 Scheduling using behavioral templates

Tai Ly, David Knapp, Ron Miller, Don MacMillen
January 1995 Proceedings of the 32nd ACM/IEEE conference on Design '95

**Publisher:** ACM Press

Full text available: pdf(69.60 Additional Information: full citation, reference KB)

KB) index terms

17 A parallel, real-time garbage collector

Perry Cheng, Guy E. Blelloch

May 2001 ACM SIGPLAN Notices, Proceedings of the ACM SIGPLA on Programming language design and implementation PLD Issue 5

**Publisher:** ACM Press

Full text available: pdf(1.82 Additional Information: full citation, abst citings, index ten

We describe a parallel, real-time garbage collector and present experime demonstrate good scalability and good real-time bounds. The collector is shared-memory multiprocessors and is based on an earlier collector algorithm was designed for simple analysis, it had some impractic paper presents the extensions necessary for a pract ...

18 Reducing virtual call overheads in a Java VM just-in-time compiler

Junpyo Lee, Byung-Sun Yang, Suhyun Kim, Kemal Ebcioğlu, Erik Altmar C. Chung, Heungbok Lee, Je Hyung Lee, Soo-Mook Moon March 2000 ACM SIGARCH Computer Architecture News, Volume 2

Publisher: ACM Press

Full text available: pdf(994.66 KB) Additional Information: full citation, abst

Java, an object-oriented language, uses *virtual methods* to support the ex classes. Unfortunately, virtual method calls affect performance and thus implementation, especially when just-in-time (JIT) compilation is done. *type feedback* are solutions used by compilers for dynamically-typed obj languages such as SELF [1, 2, 3], where virtual call overheads are much performance than in Java. Wi ...

Keywords: Java JIT compilation, adaptive compilation, inline cache, type method call

19 Using cache line coloring to perform aggressive procedure inlining

Hakan Aydin, David Kaeli

March 2000 **ACM SIGARCH Computer Architecture News**, Volume 2 **Publisher:** ACM Press

Full text available: pdf(701.54 Additional Information: full citation, abst KB) terms

Memory hierarchy performance has always been an important issue in codesign. The likelihood of a bottleneck in the memory hierarchy is increased

improvements in microprocessor performance continue to outpace those memory system. As a result, effective utilization of cache memories is es architectures. The nature of procedural software poses visibility problems perform program optimization. One approach to increasing visibil ...

20 Reducing the cost of branches by using registers

♠ Jack W. Davidson, David B. Whalley

May 1990 ACM SIGARCH Computer Architecture News, Proceeding annual international symposium on Computer Architecture 18 Issue 3a

**Publisher:** ACM Press

Full text available: pdf(1.11 Additional Information: full citation, abst citings, index ten

In an attempt to reduce the number of operand memory references, many have thirty-two or more general-purpose registers (e.g., MIPS, ARM, Sp Without special compiler optimizations, such as inlining or interprocedu allocation, it is rare that a compiler will use a majority of these registers paper explores the possibility of using some of these registers to hold braaddresses and the corresponding instruction at each branc ...

Results 1 - 20 of 200

Result page: 1 2 3 4 5 6 7 8 9 1

The ACM Portal is published by the Association for Computing Machinery ACM, Inc.

Terms of Usage Privacy Policy Code of Ethics Contact

Useful downloads: Adobe Acrobat QuickTime Windows Med Player



inline cfg

1980

2

## Scholar All articles Recent articles Results 11 - 20 of about 631 for inline c

#### **All Results**

**G** Holloway

Z Budimlic

S Sinha

K Chaki

K Kennedy

Slicing concurrent java programs - group of 4 »

Z Chen, B Xu - ACM SIGPLAN Notices, 2001 -

portal.acm.org

... sto a node s 2 ifs denotes the exit node in the CFG of

thread ... And we inline all called

methods in which synchronization methods are called, into

control flow ...

Cited by 33 - Related Articles - Web Search - BL Direct

[PS] The design and implementation of a high-performance Erlang compiler - group of 5 »

T Lindgren, C Jonsson - 1999 - docs.uu.se

... In contrast with ordinary CFGs, the Hipe CFG can have multiple entry points. ... initial

sim- plications, type analysis and type optimization, and inline expansion ...

Cited by 3 - Related Articles - View as HTML - Web Search

Impact of design of sharp non-uniform fibre Bragg gratings on system performance - group of 2 »

EG Turitsyna, JD Ania-Castanon, SK Turitsyn, L ... -

Electronics Letters, 2003 - ieeexplore.ieee.org

... suggest that, to apply approximated flat-dispersion gratings as inline tilten in ...

fibre nonlineariti and non-ideal chirped fibre gmting (CFG) liarafeiisfis iii ...

Related Articles - Web Search - BL Direct

Two-output-port fast tunable filter with low loss and low lossvariation for 32 wavelength channels - group of 2 » A Misawa, K Sasayama, T Matsunaga - Electronics Letters, 1999 - ieeexplore.ieee.org

... 80km. A satisfactory power penalty can be obtained when the CFG-OLAs are used as power and in-line amplifiers. Introduction: Chirped ...

Related Articles - Web Search - BL Direct

JaMake: A Java Compiler Environment - group of 6 »

Z Budimlic, K Kennedy - Third International Conference on Large Scale Scientific ..., 2001 - Springer

... The CFG is then passed to our assembler for bytecode generation. ... section array analysis techniques [16], it may be possible to inline heterogeneous arrays of ...

Cited by 8 Peleted Articles Web Search BL Direct

Cited by 8 - Related Articles - Web Search - BL Direct

Interactive instruction scheduling and block ordering
US Patent 6,446,258, 2002 - freepatentsonline.com
... if (Block\_empty(from)) 15 Bo\_EmptyBlock(from); 16 fi;
17 rdy.rarw.rdy-best; 18
rdy.rarw.rdy.orgate.RdySuccs(best); 19 od; 20
PathCompress(CFG); In line 1, an ...
Cited by 4 - Related Articles - Cached - Web Search

<u>Llun-a high-level debugger for generated parsers</u> - group of 4 »

S Glass, D Ince, E Fergus - Software-Practice and Experience, 2001 - doi.wiley.com

... are variations in the type of context-free grammar (CFG) that is used, the type

of parser being generated, whether the generated parser is coded inline or table ...

Cited by 2 - Related Articles - Web Search - BL Direct

On the automatic evolution of an OS kernel using temporal logic and AOP - group of 7 »

RA Aberg, JL Lawall, M Sudholt, G Muller, AF Le ... - Automated Software Engineering, 2003. Proceedings. 18th IEEE ..., 2003 - ieeexplore.ieee.org

... of the call to schedule() in line 10 of ... is a pattern to

match against CFG nodes,
RHS ... inline int wake\_up\_process(struct task\_struct \*p)
{ #ifdef CONFIG\_BOSSA ...
Cited by 22 - Related Articles - Web Search

Low-cost on-line fault detection using control flow assertions - group of 4 »

R Venkatasubramanian, JP Hayes, BT Murray - On-Line Testing Symposium, 2003. IOLTS 2003. 9th IEEE, 2003 - ieeexplore.ieee.org

... (a) (b) Figure 2. (a) A **CFG** and (b ... special assertions, for example, the assertion

in line 11 of ... instrument the C program using the inline assembly instructions ...

<u>Cited by 14 - Related Articles - Web Search</u>

Code size efficiency in global scheduling for ILP processors - group of 8 »

H Zhou, TM Conte - Interaction between Compilers and Computer Architectures, ..., 2002 - ieeexplore.ieee.org ... (b) Figure 1. (a) The **CFG** and the ... for trace- based timing simulation, the scheduled intermediate code is either converted into an **inline** execution simulator ...

<u>Cited by 8 - Related Articles - Web Search</u>

■ Goooooooooogle ▶

Result Page: <u>Previous 1 2 3 4 5 6 7 8 9 1011</u> <u>Next</u>

		2000 CO 100 CO 1
inlina afa		@aaaah
iniine cia	,	<b>BOCARCINA</b>

Google Home - About Google - About Google Scholar

©2007 Google

Home | Login | Logor



### Welcome United States Patent and Trademark Office

#### □ Search Results

BROWSE SEARCH LEEF

Results for "(((inlin\* minimiz\* i/o)<in>metadata)) <and> (pyr >= 1980

Your search matched 0 documents. A maximum of 100 results are displayed, 25 to a page, sorted by Relevance Descending order.

### » Search Options

View Session History

New Search

**Modify Search** 

(((inlin\* minimiz\* i/o) $\leq$ in>metadata))  $\leq$ and> (nyr>= 1

☐ Check to search only within this results set

» Kev

**IEEE** 

Journal or

Magazine

**IEE Journal** 

or Magazine

TEEE IEEE

Conference

Proceeding

CNF **IEE** 

Conference

Proceeding

**IEEE** 

Standard

No results were found.

**Display** © Citation ©

Please edit your search criteria and try again. Refer assistance revising your search.

Indexed by inspec\* IEEE Xplore

Home | Login | Logor

### Welcome United States Patent and Trademark Office

#### Search Results

BROWSE SEARCH LITT

Results for "(((inlin\* minimal\* i/o)<in>metadata)) <and> (pyr >= 1980Your search matched 0 documents. A maximum of 100 results are displayed, 25 to a page, sorted by Relevance Descending order.

### » Search Options

View Session History New Search

Modify Search

 $(((inlin* minimal* i/o) \le in \ge metadata)) \le and \ge (nyr \ge 1)$ 

☐ Check to search only within this results set

Format: © Citation © Citation &

» Key

**IEEE** 

Journal or Magazine

**IEE Journal** 

or Magazine

IEEE IEEE

Conference Proceeding

FFF **IEE** 

Conference Proceeding

IEEE IEEE STD Standard No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

indexed by inspec\*

Home | Login | Logor



# **Welcome United States Patent and Trademark Office**

### Search Results

BROWSE SEARCH CITTLE

Results for "(((inline i/o)<in>metadata)) <and> (pyr >= 1980 <and> pyr Your search matched 0 documents. A maximum of 100 results are displayed, 25 to a page, sorted by Relevance Descending order.

# » Search Options

<u>View Session</u> History

New Search

» Key

JEEE IEEE

Journal or

Magazine

IEE Journal

or Magazine

IEEE IEEE

Conference

Proceeding

IEE IEE

Conference

Proceeding

**IEEE** Standard Modify Search

 $(((inline i/o) \le in \ge metadata)) \le and \ge (nyr \ge 1980 \le ance)$ 

□ Check to search only within this results set

Display © Citation © Citation &

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

indexed by inspec\* IEEE Xplore

Home | Login | Logor

# Welcome United States Patent and Trademark Office

## □ Search Results

BROWSE SEARCH GUID

Results for "(((inline) <in>metadata)) <and> (pyr >= 1980 <and> pyr < Your search matched 340 of 1484991 documents. A maximum of 100 results are displayed, 25 to a page, sorted by Relevance Descending order.

w Sparch Ontions					
<ul> <li>Search Options</li> <li>View Session</li> <li>History</li> <li>New Search</li> </ul>	Modify Search  (((inline ) <in>metadata)) <and> (nyr &gt;= 1980 <and> □ Check to search only within this results set</and></and></in>				
» Key JEEE IEEE JNL Journal or	Format: Citation & Abstract				
Magazine  IEE Journal or Magazine  IEEE IEEE	1. A design of high performance inline hear Ohwe, T.; Yoneoka, S.; Aruga, K.; Yamao Magnetics, IEEE Transactions on Volume 26, Issue 5, Sep 1990 Page(s):24 Digital Object Identifier 10.1109/20.1047.				
Conference Proceeding  IEE CONF CONF CONF CONF CONF CONF CONF CONF	AbstractPlus   Full Text: PDF(292 KB) Rights and Permissions  2. Inline TM/sub 110/-mode filters with hi bypass couplings of nonresonating TE/s Rosenberg, U.; Amari, S.; Bornemann, J.; Microwaye Theory and Techniques, IEEE Volume 51, Issue 6, June 2003 Page(5): I				
IEEE Standard	AbstractPlus   References   Full Text: PDF Rights and Permissions				
	<ul> <li>3. 1.28 Tbit/s (32/spl times/43 Gbit/s) field km) DSF using L-band remotely-pumpinline amplifiers         Masuda, H.; Kawakami, H.; Kuwahara, S.         Electronics Letters         Volume 39, Issue 23, 13 Nov. 2003 Page Digital Object Identifier 10.1049/el:20031         AbstractPlus   Full Text: PDF(285 KB)</li> <li>4. Accurate synthesis of inline prototype figuadruplet sections         Macchiarella. G.:</li> </ul>				

Microwaye Theory and Techniques, IEEE Volume 50, Issue 7, July 2002 Page(s): 1 Digital Object Identifier 10.1109/TMTT.2 AbstractPlus | References | Full Text: PDF Rights and Permissions

- 5. Feed-forward continuous and complete rotatable-variable waveplate and inline Hirabayashi, K.; Amano, C.; Lightwaye Technology, Journal of Volume 21, Issue 9, Sept. 2003 Page(s): Digital Object Identifier 10.1109/JLT.200 AbstractPlus | References | Full Text: PDF Rights and Permissions
- 6. Multispan inline and adaptive group de 40 Gb/s with optical FIR-filters
  Bohn, M.; Rosenkranz, W.; Mohs, G.;
  Optical Fiber Communication Conference
  17-22 Mar. 2002, Page(s): 665 66
  Digital Object Identifier 10.1109/OFC.200
  AbstractPlus | Full Text: PDF(340 KB)
  Rights and Permissions
- 7. Bi-directional WDM transmission by us without isolators
  Jianjun Yu; Jeppesen, P.;
  Optical Fiber Communication Conference
  Volume 3, 2001 Page(s); WDD58-1-WL
  Digital Object Identifier 10.1109/OFC.200
  AbstractPlus | Full Text: PDF(272 KB)
  Rights and Permissions
- 8. Thermofluid analysis of staggered and in Dvinsky, A.; Bar-Cohen, A.; Strelets, M.; Thermal and Thermomechanical Phenome ITHERM 2000. The Seventh Intersociety Volume 1..23-26 May 2000 Page(s): Digital Object Identifier 10.1109/ITHERN AbstractPlus | Full Text: PDF(1352 KB) Rights and Permissions
- 9. Inline network encryption for multimed Aura Ganz; Se Hyun Park; Ganz, Z.: Military Communications Conference, 19: LEE
  Volume 2, 18-21 Oct. 1998 Page(s):560 Digital Object Identifier 10.1109/MILCO AbstractPlus | Full Text: PDF(420 KB) Rights and Permissions
- 10. Scale model experiments on optimum tor resolution of electromagnetic responsaking, S.T.; Gupta, O.P.; Negi, J.G.; Geoscience and Remote Sensing, IEEE T Volume 28, Issue 6, Nov. 1990 Page(s): Digital Object Identifier 10.1109/36.6262

  AbstractPlus | Full Text: PDF(308 KB) Rights and Permissions

- 11. Inline capacitive and DC-contact MEM Muldavin, J.B.; Rebelz, G.M.; Microwave and Wireless Components Le Microwave and Guided Waye Lefters Volume 11, Issue 8, Aug. 2001 Page(s): Digital Object Identifier 10.1109/7260.94

  AbstractPlus | References | Full Text: PD Rights and Permissions
- 12. Equipment productivity improvement implementation
  Lafferty, N.; Fiol, B.; Jowett, P.; Karzhav Advanced Semiconductor Manufacturing Workshop
  30 April-2 May 2002 Page(s):218 222
  Digital Object Identifier 10.1109/ASMC.
  AbstractPlus | Full Text: PDF(589 KB)
  Rights and Permissions
- 13. ViaSat Internet Protocol Crypto inline Internet
  Agnew, M.; Goodwin, J.; Quintana, R.; Military Communications Conference Profile Editory Communications Conference Profile Volume 1, 31 Oct.-3 Nov. 1999 Page(s): Digital Object Identifier 10.1109/MILCO AbstractPlus | Full Text: PDF(496 KB) Rights and Permissions
- 14. 40-Gb/s RZ transmission over transoce managed standard liber using a new in method Sahara, A.: Inui, T.: Komukai, T.: Kubota Photonics Technology Letters, IEEE Yolume 12, Issue 6, June 2000 Page(s): Digital Object Identifier 10.1109/68.8490 AbstractPlus | References | Full Text: PD Rights and Permissions
- 15. Polarization independent, all-fiber pha fiber DFB lasers
  Yamashita, S.; Set, S.Y.; Laming, R.I.; Photonics Technology Letters, IEEE
  Volume 10, Issue 10, Oct, 1998 Page(s): Digital Object Identifier 10.1109/68.7202
  AbstractPlus | References | Full Text: PD |
  Rights and Permissions
- 16. 40-Gb/s RZ transmission over a transomanaged standard fiber using a modified method
  Sahara, A.; Inui, T.; Komukai, T.; Kubota Lightwave Technology, Journal of Volume 18, Issue 10, Oct. 2000 Page(s). Digital Object Identifier 10.1109/50.8871
  AbstractPlus | References | Full Text: PD Rights and Permissions

- 17. 3.08 Tbit/s (77×42.7 Gbit/s) WDM tran 100 km repeater spacing using dual C-Raman/erbium-doped inline amplifiers Zhu, B.; Leng, L.; Nelson, L.E.; Olan, Y. Stulz, L.; Chandrasekhar, S.; Radic, S.; V Feder, K.S.; Thiele, H.; Bromage, J.; Gruelectronics Letters Volume 37, Issue 13, 21 June 2001 Page Digital Object Identifier 10.1049/el:20070 AbstractPlus | Full Text: PDF(212 KB)
- 18. Fibre optic modulators using active mu Johnstone, W.; Murray, S.; Thursby, G.; Culshaw, B.; Electronics Letters Volume 27. Issue 11. 23 May 1991 Page AbstractPlus | Full Text: PDF(264 KB)
- 19. Yield prediction using critical area and Zhou, C.; Ross, R.; Vickery, C.; Metteer, Advanced Semiconductor Manufacturing Workshop
  30 April-2 May 2002 Page(s):82 86
  Digital Object Identifier 10.1109/ASMC.
  AbstractPlus | Full Text: PDF(497 KB)
  Rights and Permissions
- 20. The economic impact of choosing off-lindeployment in semiconductor manufact Spanos, C.J.; Jula, P.; Leachman, R.C.; Semiconductor Manufacturing Symposium 8-10 Oct. 2001 Page(s):37-40 Digital Object Identifier 10.1109/ISSM.29 AbstractPlus | Full Text: PDF(336 KB) Rights and Permissions
- 21. Inline monitoring of multi-level dual in technologies
  Kolagunta, V.; Smith, B.; Islam, R.; Angy S.; Duraiswami, N.; Jana, P.; Veeraragha' Interconnect Technology Conference, 200 International 5-7 June 2000 Page(s):247 249
  Digital Object Identifier 10.1109/IITC.20
  AbstractPlus | Full Text: PDF(244 KB)
  Rights and Permissions
- 22. A highly miniaturized recursive Z-path Freisleben, S.; Bergmann, A.; Bauernschi Ultrasonics Symposium, 1999. Proceedin Volume 1, 17-20 Oct. 1999 Page(s):347 Digital Object Identifier 10.1109/ULTSY AbstractPlus | Full Text: PDF(316 KB) Rights and Permissions
- 23. High bitrate operation of a novel optication of a novel optication of a novel optication of a novel optication of a novel optical communication of a novel optical communication of a novel optical optical communication of a novel optical optical communication of a novel optical optical optical communication of a novel optical opt

Volume 1, 20-24 Sept. 1998 Page(s):183 Digital Object Identifier 10.1109/ECOC. AbstractPlus | Full Text: PDF(184 KB) Rights and Permissions

- 24. Fine-grain software distributed shared Scales, D.J.; Gharachorloo, K.: Aggarwal High-Performance Computer Architecture International Symposium on 1-4 Feb. 1998 Page(s):125 136 Digital Object Identifier 10.1109/HPCA.: AbstractPlus | Full Text: PDF(132 KB) Rights and Permissions
- 25. High frequency STW resonator filters Almar, R.; Horine, B.; Andersen, J.; Ultrasonics Symposium, 1992, Proceedin 20-23 Oct. 1992 Page(s):51 56 vol. Digital Object Identifier 10.1109/ULTSY AbstractPlus | Full Text: PDF(392 KB) Rights and Permissions

ज्ञ Inspec"

IEEE Xplore®

Home | Login | Logoi

# Welcome United States Patent and Trademark Office

### Search Results

BROWSE SEARCH GUID

Results for "(((code expansion optimization )<in>metadata)) <and> (py <and> pyr &l...</a> Your search matched 0 documents.
A maximum of 100 results are displayed, 25 to a page, sorted by Relevance Descending order.

# » Search Options

View Session History

New Search

Modify Search

ردode expansion ontimization\<in>metadata\) <anc

□ Check to search only within this results set

Format: © Citation © Citation & Abstract

» Key

HEEE IEEE

Journal or

Magazine

IEE Journal

or Magazine

IEEE IEEE

Conference

Proceeding

LEE IEE

Conference

Proceeding

IEEE STD

**IEEE** 

Standard

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

ज्य Inspec

IEEE Xplore

Home | Login | Logor

# Welcome United States Patent and **Trademark Office**

### Search Results

BROWSE SEARCH FIRE

Results for "(((code expansion dependence)<in>metadata)) <and> (pyr <and> pyr <= ... in the control of the cont

# » Search Options

View Session History

New Search

» Kev

**IEEE** 

Journal or Magazine

**IEE Journal** 

or Magazine

TEEE IEEE

Conference Proceeding

CNF **IEE** 

Conference Proceeding

IEEE IEEE STD Standard

## **Modify Search**

(((code expansion dependence)<in>metadata)) <ani

☐ Check to search only within this results set

Format: © Citation © Citation & Abstract

## No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

Indexed by inspec\*



inline I/O

1980

2

# Scholar All articles Recent articles Results 1 - 10 of about 17,500 for inline

### **All Results**

V Pai

A Fetting

P Druschel

W Zwaenepoel

M Berry

<u>IO-Lite: a unified I/O buffering and caching system - group of 16 »</u>

VS Pai, P Druschel, W Zwaenepoel - ACM Transactions on Computer Systems (TOCS), 2000 - portal.acm.org ... IO-Lite: A Unified I/O Buffering and Caching System • 47 ... Small data items such as

network packet headers are still stored inline in mbufs, but the performance ...

Cited by 202 - Related Articles - Web Search - BL Direct

Programmable I/O device identification - group of 2 »

JM McVey - US Patent 4,360,870, 1982 - Google Patents
... source may, for instance, be a dual inline switch
setting ... I/O devices being applied
first to the highest ... line 51, the word "incuding" should
read —including ...
Cited by 61 - Related Articles - Web Search

[BOOK] Techniques for efficient inline tracing on a shared-memory multiprocessor - group of 3 »

SJ Eggers, DR Keppel, EJ Koldinger, HM Levy - 1990 - ACM Press New York, NY, USA

... The basic approach, **inline** tracing, has been used in other tracing and performance

tools. ... Section 4 discusses the buffering of trace data for I/O. Sec- tion 5 ...

<u>Cited by 93</u> - <u>Related Articles</u> - <u>Web Search</u> - <u>Library</u> <u>Search</u>

Hierarchical modeling and analysis of embedded systems - group of 9 »

R Alur, T Dang, J Esposito, Y Hur, F Ivancic, V ... -

Proceedings of the IEEE, 2003 - ieeexplore.ieee.org ... Although modularity in hybrid specifications has been addressed in languages such as hybrid input—output (I/O) automata [14], CHARON allows richer ...

<u>Cited by 65</u> - <u>Related Articles</u> - <u>Web Search</u> - <u>BL Direct</u>

... extension software containing a self-describing feature table for accessing I/O devices according to ... - group of 6 »

A Sato, DC Baker, CJ Waldron - US Patent 5,291,585, 1994 - Google Patents

... [54] COMPUTER SYSTEM HAVING SYSTEM FEATURE EXTENSION SOFTWARE CONTAINING A

SELF-DESCRIBING FEATURE TABLE FOR ACCESSING I/O DEVICES ACCORDING TO ... in I/O devices. ...

Cited by 124 - Related Articles - Web Search

Universal I/O pad structure for in-line or staggered wire bonding or arrayed flip-chip assembly - group of 3 » SJ Bassett - US Patent 6,242,814, 2001 - Google Patents ... The present invention utilizes multiple rows, in depth, ofbond sites placed either staggered or in-line on the I/O cells. Common ... Cited by 11 - Related Articles - Web Search

I/O riser card for motherboard in a personal computer/server - group of 2 »

LC Cobb, GM Kuzmanich - US Patent 5,519,573, 1996 - Google Patents
... is secured to the chassis in parallel with the bottompanel. Themotherboardhas asocket, such as a single in-line memory module (SIMM) socket. The I/O riser card ...

Cited by 19 - Related Articles - Web Search

Jaguar: enabling efficient communication and I/O in Java

- group of 18 »

M Welsh, D Culler - Concurrency - Practice and Experience, 2000 - doi.wiley.com

... Jaguar: enabling efficient communication and I/O in Java ... 520 M. WELSH AND D. CULLER environment, high-performance communication and I/O play a dominant role. ...

<u>Cited by 64</u> - <u>Related Articles</u> - <u>Web Search</u> - <u>BL Direct</u>

System for interconnecting I/O modules for data communications over a common backplane - group of 2 » P Nordenstrom, RJ Coppenhaver - US Patent 5,472,347, 1995 - Google Patents

... multiple I/O modules installed in an inline fashion in ... on an adjacent base unit of another I/O block unit ... units 14 are mounted in adjacent in line positions on ...

<u>Cited by 16</u> - <u>Related Articles</u> - <u>Web Search</u>

Epigenetic variants of a transgenic petunia line show hypermethylation in transgene DNA: an ... - group of 4 » P Meyer, I Heidmann - Molecular Genetics and Genomics, 1994 - Springer

... intensively pigmented line, line II showed complete pigmentation with slightly reduced intensity, line III displayed a marbled colouration and in line IV no Al ...

<u>Cited by 49 - Related Articles - Web Search - BL Direct</u>

Goooooooogle >

Result Page: 1 2 3 4 5 6 7 8 9 10 Next

	 		The same of the sa		4.0
inline I/O			Sea	alti	

Google Home - About Google - About Google Scholar

©2007 Google